cgunders@nps.edu

831 224 5182

AUTOMATED JCIDS MANUAL

Encl: (1) SIDECAR Executive Summary

(2) SIDECAR Executive Briefing

Background: Defense Enterprise (DE) Information Technology (IT) acquisition is broken, per myriad watchdog reports. 2010 Nat'l Defense Authorization Act Sect 804 mandates that DoD fix IT acquisition. Einstein observed that trying to fix a problem with the process that created it is madness.

Discussion:

The DE acquisition documentation process described in the JCIDS manual is implemented via an artisan process that generates a serial progression of long, expensive, static documents.

The various governance and requirements documents that drive the artisan DE acquisition process typically provide subjective and frequently conflicting guidance.

Multiple programs generate duplicative documentation.

Adding more bureaucracy will not streamline any process or improve oversight.

To address these issues, the Marine Corps Combat Development Center (MCCDC), has developed a prototype of an automated approach for executing JCIDS Manual guidance called ***s***emantically ***i***nformed ***d***ynamic ***e***ngineering of ***c***apabilities ***a***nd ***r***equirements (SIDECAR).

SIDECAR uses a "TurboTax" metaphor, i.e. a simple user interface links to complex semantically interoperable databases of objective policies, requirements, technologies, etc. and automatically generates whatever document, e.g. CDD, CPD, ISP, TEMP, etc are required.

SIDECAR is being developed in partnership with the JCS Capabilities Development Tracking Management (CDTM) system.

Recommendation: Observe SIDECAR demo.

Enclosure 1: Executive Summary of SIDECAR: Semantically-Informed Dynamic Engineering of Capabilities And Requirements

Executive Summary of SIDECAR: Semantically-Informed Dynamic Engineering of Capabilities And Requirements

Motivation. There are serious size, speed, and security (S^3) problems in the Defense Community fielding a cohesive Global Information Grid, and in the Acquisitions and Requirements Communities producing and maintaining up to date CDD/ICD models of each program individually let alone their interdependencies. Those problems can't be avoided, since effective shared information access is already a key factor in warfare, and C3I is already heavily dependent on computer networks.

But looming just *beyond* S^3 is an even higher peak: *cognitive capability,* the bottleneck caused by the lack of semantic depth of understanding with which our software processes information. That lack of semantic depth manifests, e.g., in the brittle way DODAF (Dept. of Defense Architecture Framework) formats display our architecture products and the voluminous amount of statutory, regulatory, and prescriptive policies and directives emanating out of all DOD entities from engaged MAGTFs up through the Offices of the Secretary of Defense. It would be absurd to imagine the intelligence community employing human idiot savants in all the crucial roles of (everything except the very last stage of) processing information up through high-level predictive analysis, and yet we blithely rely on brittle *software* idiot savants in exactly those crucial roles!

Thanks to recent progress in automated reasoning, we can now do something about this problem.

The task. Consider a program serving as an intelligent auxiliary to the requirements officer assigned to produce a Capability Development Document (CDD) for a target system, say DCGS-MC. This task is reminiscent of filling out a complex income tax return (e.g., having to cope with legacy and changing regulations, policies, and directives), but in some ways much more complicated because the to-be-developed target system must be properly linked to all other relevant existing information/communications system architectures as represented in their requirements documentation (e.g. CDDs and CPDs).

SIDECAR is a pilot project performed in 2010, under USMC direction, to demonstrate this capability. The contractor, Cycorp (www.cyc.com), was able to execute this as a small ($300k) effort by leveraging its enormous already-existing ontology, knowledge base, natural language dialogue and inference engine technologies – collectively called Cyc – extending each of these components only as needed. SIDECAR starts with partial information about relevant systems such as MCISR-E and organizations such as JROC, and a model of what information goes into which sections of a CDD (i.e., a declarative representation of which set

of questions' answers *comprise* each section of a capability development document). SIDECAR uses that model to drive a mixed-initiative clarification dialogue, with its user, to obtain more and more information – much as TurboTax does in *its* domain. In this case, SIDECAR learns more and more about the target system, DCGS-MC. Having a *CDD model* means that all this new information is relevant to, and has a well-understood place in, the CDD. Although the dialogue is in English, the internal model in SIDECAR is represented in formal logic (n^{th}-order predicate calculus). Using Cyc's natural language generation capability, each of these assertions is converted into English, and SIDECAR's CDD model guides it in placing those English sentences in the correct order to generate the actual English CDD.[1] This happens in real time, incrementally, so the growing state of the English CDD can be viewed at any time. But instead of editing *it,* if something needs to be changed, the user is pointed back to whatever SIDECAR menu or question led to that particular English sentence or table entry. They modify their choice on that SIDECAR screen, and the English CDD updates.

SIDECAR partially *understands* what it's doing, in the sense that it can use its partially-completed model of the CDD-in-progress to infer such things as: (1) Is there a contradiction between what the user just told me and everything else that's already known? (2) Given what the user just told me, can I guess at any other questions' answers? If not, can I at least eliminate some of the possible answers to some of the questions? Are some *entire questions* now completely moot? (3) At any given moment, where should the user best spend their effort, and focus their attention, next? (4) If some of the alternative design choices have been left unspecified by the architects, what are the pros/cons of each alternative? In the current pilot project version of SIDECAR, all 4 of these types of beneficial inference already clearly occur.

Future capabilities:

- Capability (4), above, could be expanded, producing a version that the software architects could use in designing the programs prior to the documentation phase.
- SIDECAR could be utilized in training new Acquisitions personnel.
- Regression testing against policy changes: rerun each program's design automatically every day, and auto-alert if new regulations – or changes in technology capabilities, vulnerabilities, or costs – or " real world" changes

[1] Instead of producing the final English CDD itself, SIDECAR could at this point hand the English sentences and paragraphs to CDTM to assemble and archive. Even in this case, the declarative logical SIDECAR model would be archived and associated with this CDD, much like a TurboTax **.tax** file, so a user could later pick up editing that CDD using SIDECAR.

(e.g., political boundaries or conflicts) make some system's detailed design now incompatible with its intended capabilities.

- Instead of just coping with those changes *after* they occur, SIDECAR could be used to answer "what if" questions of the models for any or all of these systems, to help analyze the impact of possible changes in policy, technology, etc. which are not yet decided.
- Represent existing systems and the terrain of regulations and policies in which they live. Those semantic models would jointly enable reasoning about FOS/SOS dependencies to be more complete and machine-reasonable, leading SIDECAR to make increasingly correct suggestions about what the new target system should link to (and how and why) or incorporate as a component, and suggestions about design alternatives' pros/cons.
- Extend that to interoperate with SIGINT, ELINT, HUMINT,… data represented in Cyc to detect attacks earlier and to generate populations of plausible threat scenarios which in turn could inform changes in policy, information collection, and operations.

Conclusion. Looking beyond raw speed, scale, and security bottlenecks, some action officers at Marine Corps Combat Development Command recognize the importance of developing a suite of machine-readable – and deeply machine-*reason*able – acquisition documentation which is deliberately focused on content rather than form, and which explicitly represents enterprise goals and the relation of program elements to those goals. SIDECAR demonstrates the feasibility of that vision, and exhibits several forms of TurboTax-like help that would be most welcome in practice.